製品ライフサイクル高信頼化―仕様と実装と環境のギャップをライフサイクルで管理する技術

Advanced Technologies for Dependable Systems through Product Life Cycle by Managing Gaps among Specifications, Implementation, and Environment

内平　直志

■ UCHIHIRA Naoshi

製品の大規模化，複合化，オープン化，及び協働化に伴い，設計・製造時には想定できない様々な状況変化に対しても，製品ライフサイクルを通じて高い品質を維持することが求められている。

東芝は，ライフサイクルの中で仕様，実装，あるいは環境の変化によって生じるそれぞれの間のギャップを品質の劣化ととらえ，そのギャップを監視，検出，及び修正することで高信頼化を実現する“製品ライフサイクル高信頼化モデル”に基づき，製品を構成するハードウェアやソフトウェアだけでなく，人間系プロセスを含むシステム全体の高信頼化を実現する技術開発を行っている。

With the increasing scale, complexity, open and cooperative operability of products, the conventional quality assurance approach is insufficient to respond to the expanding gaps among their specifications, implementation, and environment.　New approaches are therefore required to establish high dependability by monitoring, detecting, and modifying these gaps throughout the product life cycle.

Toshiba is developing advanced technologies baced on the life cycle quality assurance model for highly dependable products.

製品の高信頼化への時代の要請と実現の難しさ

電子情報通信技術の進展による製品の大規模化，複合化，オープン化，及び協働化に伴い，社会，顧客，企業からの製品の信頼性に対する要請はますます強まっている。

例えば，社会インフラを構成する製品は，互いに密に連携しながら協働しており，一つの事故や故障が社会的に重大な影響を及ぼすケースが増えている[1]。同時に，安心や安全に対する顧客の意識が高まっており，安心・安全社会に対する企業の社会的責任も増している。産業界でも，宇宙航空や自動車，鉄道，エレベーターなどの分野では，製品の信頼性を実現するプロセスを規定した“機能安全”（IEC 61508：国際電気標準会議規格61508）などの規格化が進んでいる。

一方，コモディティ製品（日用品化した製品）でも，故障や不具合の修理，交換などの品質コスト管理は，企業の重要な経営課題である。とりわけ，製品のターゲットがグローバルかつボリュームマーケットである場合は，数量のメリットを追求する分，品質コストは大きなリスクとなっている。

これらの高信頼化への要請に対して，その実現は以下の2点で容易ではない。

(1)　製品が稼働する外部環境に対してオープン・協働化してきており，製品使用時，保守時，更新時などの製品ライフサイクルにおける様々な状況変化を，設計・製造時にすべて想定して品質管理を行うことは難しい。

(2)　製品がハードウェア（HW）やソフトウェア（SW）だけでなく人間系のプロセスを含めて複合化が進んでおり，従来型の故障と原因の線形的な因果関係に基づいて個々の要素の品質管理を積み上げるだけでは，システム全体の信頼性を確保することは難しい[2]。

製品の品質（Q：Quality），コスト（C：Cost），納期（D：Delivery）は，それぞれトレードオフ（二律背反）の関係にある。高い品質を実現するために，無制限の期間とコストを掛けることは許されない。更に，単なるQCDの達成だけでなく，実際に製品を使う段階での顧客の満足度や顧客企業の価値を向上させることが求められている。これらのトレードオフの中で最適な信頼性を達成するためには，従来型の製品設計・製造時の品質管理に加えて，製品ライフサイクルを通じた品質管理が不可欠である。

製品ライフサイクルでの仕様と実装と環境のギャップ

ここでは，製品の品質の劣化を仕様と実装と環境のギャップとして統一的にとらえる。ここで，仕様とは，設計者が意図した製品が持つべき機能や性能であり，実装とは，仕様をHW，SW，及び人間系プロセスで実現したものであり，環境とは，製品が稼働する外部環境であるとともに，製品の価値を享受するユーザーの使用状況も含む。

以下に，製品の仕様と実装と環境のギャップと品質の劣化の関係について述べる。HW，SW，及び人間系プロセ

スにおける品質を，仕様と実装と環境のギャップとして統一的にとらえることで，製品ライフサイクルを通じたシステム全体としての高信頼化を見通しよく考えることができる。

■仕様と実装のギャップ

HWの場合は，物理的に経年劣化したHW（実装）が当初の機能や性能（仕様）を満たせなくなると，ギャップが生じる。HWの経年劣化は必ず発生するため，高信頼性が求められる製品では，定常的に仕様と実装のギャップを監視し，部品交換などでギャップを解消しなければならない。

SWの場合，設計・製造時に仕様どおりに実装（プログラム）ができていない“バグ”はギャップの典型であり，出荷前にコストを掛けて検査や検証を行うが，現実には完全にバグを除去するのは難しく，出荷後に実装の更新（ギャップの解消）が行われることも珍しくない。また，保守時の実装の修正により，徐々に仕様と実装が一致しなくなり，そのギャップが保守を難しくしている。

人間系のプロセスでも，現場での運用上の変更が仕様に反映されていない場合には仕様と実装のギャップが生じ，その累積により事故に至るケースもある。

■仕様と環境のギャップ

仕様と環境のギャップは，ユーザーが仕様では想定しなかった製品の使い方をしている場合に発生する。また，製品の稼働環境が仕様で想定していたものから変わってしまった場合もギャップが生じる。Webブラウザをバージョンアップしたため，Webアプリケーションが動かなくなる場合などは，その例である。

■実装と環境のギャップ

実装と環境のギャップとしては，従来は正常に稼働していたものが，外部環境の変化により，実装上の許容範囲（設計マージン）を超えてしまう，あるいは微妙なインタフェースやタイミングの不一致が発生するなどにより不具合が生じることがある。特に，設計マージンは暗黙的にチューニングされていることも多く，不具合を事前にチェックすることは難しい。

製品ライフサイクル高信頼化モデル

仕様，実装，及び環境の経年変化は必然であり，ギャップの存在自体は“あってはならないもの”ではなく，不可避なものとして対処する必要がある。そこで，ギャップを可能な限り早期に検出し，早期に対策し，影響を局所化することが求められる。

HWに関しては，実装の劣化に対する保守が不可欠であり，従来からそのような品質保全が行われていた。東芝は，HWだけでなく，SWや人間系プロセスにまで劣化（ギャップの拡大）の概念を拡張するとともに（**囲み記事参照**），仕様，実装，及び環境は変化するものであるという立場から，製品ライフサイクルを通じたギャップの監視，検出，修正により高信頼化を実現する“製品ライフサイクル高信頼化モデル”を提案している（**図1**）。

前述のように，設計・製造時の品質管理にコストをいくら掛けても，製品ライフサイクルを通じた品質保全には限界があり，ライフサイクル高信頼化への発想の転換は不可避であろう。ライフサイクル高信頼化への変革には，それを実現するための仕組みを構築する投資が必要であり，イニシャルコストは増大するかもしれないが，ライフサイクルコストでは投資に見合うコスト削減効果があると思われる。

また，ライフサイクル高信頼化は顧客との継続的な関係を前提としており，製造業のサービス事業化と位置づけることができる[3]。更に，使い捨てからリデュース（延命化），リユース（再利用），及びリサイクル（再資源化）への移行を促すことになり，地球環境に対しても有益である。

製品ライフサイクル高信頼化を実現する技術

ライフサイクル高信頼化の実現には，それを可能にする技術革新が不可欠であるが，技術面でも以下に述べる四つの変化点を迎えており，製品の品質保全におけるイノベーションの準備は整いつつある。

(1)　製品がネットワークに接続され，製品の使用状況のオンライン監視が

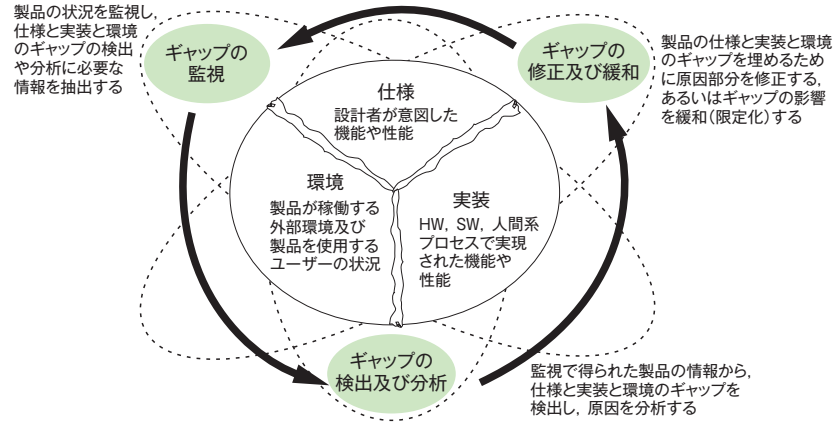


図1．製品ライフサイクル高信頼化モデル ― 製品の仕様，実装，及び環境間で経年的に拡大するギャップを，監視，検出及び分析，修正及び緩和のサイクルで高信頼化を実現する。

Product life-cycle quality assurance model

SWのライフサイクルにおける仕様と実装と環境のギャップ

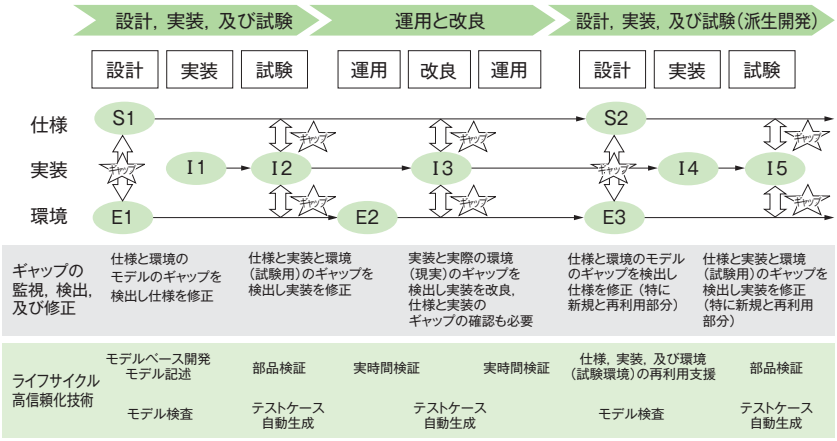
SWは，HWのように実装自体が経年劣化することはないが，外部環境の変化や継続的な保守や改良，更には派生開発により，仕様，実装，及び環境にギャップが生じ，品質が劣化する。図では，モデルベース開発を前提として，“設計，実装，及び試験”，“運用と改良”，“設計，実装，及び試験(派生開発)”の三つの段階で，仕様と環境と実装のギャップに対する監視，検出，及び修正のサイクルを示している。

最初の設計，実装，及び試験の段階では，仕様と環境のモデルのギャップをモデル検査などで検出するとともに，実装と仕様及び環境のギャップを部品検証やテストケースの自動生成などで検出し修正する。ここでは，従来型のV字モデルによる高信頼化手法が適用されるが，SWリリース後の運用時には当初想定していた外部環境が変化する場合がある。

運用と改良の段階では，実装と環境のギャップを実時間検証などにより検出し改良する。ここで，実装の改良後の確認(リグレッションテスト)にテストケース自動生成が効果を発揮する。

更に，派生開発段階では，仕様や実装の新規開発部分と再利用部分のギャップを検出し解消する。

このように，モデルベース開発の中で，仕様と実装と環境のギャップを継続的に検出し解消することにより，ライフサイクルコストが低減する。ここに，モデルベース開発及び形式手法に基づくテスト検証のQCDでの優位性が出てくる。



S：Specification(仕様)　I：Implementation(実装)　E：Environment(環境)

可能になってきた。同時に，センサ技術が進化し，コモディティ製品にもセンサが付くようになった。また，LSIの進化により，稼働状況を監視し制御する機構も低コストで実現できるようになってきた。

(2) HWの信頼性解析技術が，従来の確定論的な解析から統計論的な解析にシフトし，信頼性のリスクを定量的に扱えるようになった。また，各種部品の物理現象に対する信頼性データベースの整備が進んできた。これらにより，リスクベースメンテナンス(注1)の対象範囲が広がってきた。

(3) モデル検査法などSW検証分野の革新的な技術の開発と計算機能力の向上により，高度なシステムの解析や検証が設計・試験時だけでなく運用・実行時にも可能になってきた。

(4) システム全体あるいはプロセスに対するリスクの分析・評価手法などの安全に関する方法論が確立するとともに，システム設計や検査手法との統合が進んできた。

これらの変化点を踏まえて，製品ライフサイクル高信頼化を実現するためのコア技術について，以下に述べる(**表1**)。

ギャップ監視の支援技術

製品の状況を常時監視し，仕様と実装と環境のギャップ検出に必要な情報を抽出する。具体的には，機器の状況を遠隔から監視するセンサ技術，及び情報収集・蓄積技術がある。ここで，遠隔監視で収集可能になった膨大なデータをどのように効率的に管理するかは大きな課題であり，情報フィルタリング技術や情報圧縮技術は不可欠である。また，ここでは，プライバシーや情報セキュリティの管理も課題である。SWの実行状況の監視には，外部から監視する場合と，プログラム内部に監視機能を埋め込む場合がある。すべての実行状況を監視するのはオーバヘッド(監視のための付随処理)が大きいため，監視のチェックポイントとチェック内容をアサーション(事前条件，事後条件，及び不変条件)として定義する。これらは，“契約による設計”として具体的な技術開発が進んでおり，ギャップ監視及び次節で述べる検出の重要技術である。

ギャップ検出及び分析の支援技術

監視して得られた情報を分析し，製品の仕様と実装と環境のギャップを検出する。ギャップの検出には，設計者が設定した前提や条件を満たしているかのチェック(違反検出)と，正常な挙動を

(注1) リスクベース メンテナンス
故障の発生確率と，発生した場合の影響の大きさなどのリスクに基づき，最適なメンテナンスを行う手法。

逸脱しているかのチェック（異常検出）がある。

HWに関しては，従来の統計的品質管理に加えて，製品の状態に基づき予測保全を行うプログノスティクス＆ヘルスマネジメント（PHM：Prognostics & Health Management）技術[4]（**囲み記事参照**）が，近年注目を浴びている。PHMを実現するための要素技術として，信頼性予測技術（統計的疲労寿命予測技術，強度データベース），不具合現象のモデリング技術，大規模シミュレーション技術（大規模応力解析，連成解析，モンテカルロ解析），加速試験技術，及びデータ・テキストマイニング技術[注2]がある。

SWに関しては，形式手法（Formal Method）により仕様と実装のギャップを検出する検証技術が長年研究されてきたが，近年の計算機能力の向上とモデル検査法などの検証技術の進展により，実適用の例が増えている[5]。SWの実行時に検証（違反検出や異常検出）を行う実行時検証（Runtime Verification）[6], [7]（**囲み記事参照**）の研究開発も進んでいる。使用環境がオープンでダイナミックな場合，設計時に完ぺきな検証をすることは非現実的であり，実行時検証は重要である。仕様が不十分な場合でも，SWが安全に実行された実行履歴から，SWが通常満たしている条件（不変条件，実行順序）を抽出し[8]，実行時検証に活用する方法も有望である。また，仕様（モデル）からテストケースを生成するモデルベーステストも，仕様と実装のギャップの検出に効果が大きい。

更に，伝統的なFTA（Fault Tree Analysis）やFMEA（Failure Mode and Effects Analysis）などのリスクの分析・評価手法も，HWから人間系のプロセスを含むシステム全体への適用が進んでいる[2]。特に，人間系プロセスに関しては，ヒューマンファクター分析技術が重要となる。

表1．仕様と実装と環境のギャップに対する製品ライフサイクル高信頼化技術

Product life-cycle quality assurance technologies for managing gaps among specifications, implementation, and environment

項　目	HW	SW	人間系プロセス
ギャップ監視技術	・ユビキタスセンサネットワーク ・情報フィルタリング・圧縮 ・プライバシー・セキュリティ管理 ・保守統合データベース	・SWメトリクス ・アサーション（事前条件，事後条件，不変条件）	・業務プロセスモデリング ・インシデント情報収集システム
ギャップ検出及び分析技術	・診断自動化技術（データマイニングやエキスパートシステムによる異常検出，故障予測，原因推定） ・信頼性予測技術（統計的疲労寿命予測技術，強度データベース） ・不具合現象のモデリング ・大規模シミュレーション技術（大規模応力解析，連成解析，モンテカルロ解析） ・加速試験技術 ・統計的品質管理	・形式手法・検証（モデル検査，定理証明，データフロー解析） ・モデルベース開発と試験（テストケース自動生成，リグレッション試験） ・ソースコード差分解析 ・ソースコードからの仕様抽出 ・実行時検証（アサーションベース検証，異常検出） ・協調シミュレーション	・リスク分析，評価，及び可視化技術（FTA，FMEA，RFMEA） ・ヒューマンファクター分析 ・システム理論に基づく事故モデル ・失敗事例集
ギャップ修正及び緩和技術	・安全保護回路 ・フォルトトレラント技術（多重化など） ・スーパーバイザ制御 ・適応制御	・保護OS（メモリ保護，時間保護） ・モニタリング指向プログラミング ・適用型SW	・PDCAサイクル支援 ・ナレッジデータベース

OS：基本SW　　RFMEA：Risk FMEA　　PDCA：Plan-Do-Check-Act

プログノスティクス＆ヘルスマネジメント

プログノスティクスとは，製品や部品に対して，現在の状態を監視して将来の状態や故障の発生と場所を予測する技術である。故障が発生してから診断するダイアグノスティクス（Diagnostics）が事後保全に使われるのに対し，製品や部品の状態を考慮したコンディションベースの予測保全で使われる。

プログノスティクスに基づく製品のライフサイクル管理をプログノスティクス＆ヘルスマネジメント（PHM）と呼ぶ。

PHMは，従来から航空宇宙などのミッションクリティカルなシステムを対象に研究開発が行われてきたが，近年PCや自動車などの民生用機器にも適用されるようになってきた。

2008年10月には米国コロラド州デンバー市でPHMの初の国際会議が開催されるなど，今後の発展が期待される。

実行時検証

SWの実行時にSWの挙動が仕様を満たしているかを検証する技術であり，SWのヘルスマネジメント技術と位置づけることができる。仕様は，時相論理やオートマトンなどの形式記述で与えられる。実行時検証は，実際の稼働環境で実行された部分がチェックの対象であり，形式検証より網羅性の面では弱いが，稼働環境がオープンでダイナミックに変化する場合など，形式検証時に外部環境の挙動が完全に把握できない場合では強みを発揮する。

2001年からは実行時検証の国際ワークショップが毎年開催されており，検証技術の新しい潮流として注目されている。

（注2）　データ・テキストマイニング技術
大量のデータやテキスト中に内在する共通性や因果関係を分析し，故障分析や故障予測などの有用な知識を獲得する技術。

■ギャップ修正及び緩和の支援技術

製品やシステムの仕様，実装，及び環境のギャップが検出できた場合，そのギャップを埋める必要がある。基本的には，ギャップを検出した場合には，人間系（設計・保守担当者）が設計変更，部品交換，使用制限など対応する必要があり，そのための支援技術が必要である。場合によっては，顧客にギャップを埋めてもらうための遠隔ガイド技術も必要である。

ギャップ修正の自動化に関しては，実装とは独立に安全保護系を用意し，実装や環境の挙動が想定した範囲を逸脱した場合は，運用・実行時にその挙動を修正し，ギャップを緩和するアプローチ，例えば，再実行する，安全に停止する，実行順序を変えるなどがある。この技術は，フォルトトレラントやスーパーバイザ制御として古くから研究がされているが，ギャップの状況に，より適応して制御する技術の開発が求められている。

東芝の取組み

製品ライフサイクル高信頼化における当社の取組みを仕様と実装と環境のギャップモデルの視点から述べる（**表2**）。

■デジタル機器のヘルスモニタリング

パソコン（PC）などのデジタル機器において，埋め込まれているセンサ情報から，ユーザー使用時の性能低下や電子基板の疲労度合いといったHWの健康状態を把握し，仕様と実装の劣化状況や使用環境とのギャップを検出しユーザーに提示するとともに保守に活用する。

設計時の詳細なシミュレーション結果を生かし，モバイル機器用にセンサを極力少なくできるPHM技術が特長である（この特集のp.8－11参照）。

■プロアクティブ(注3)な品質保全

コモディティ製品では，出荷後の市場品質（修理率など）の管理が重要である。そこで，PCをモチーフに，市場品質情報から仕様（想定修理率）と実装（実際の修理率）のギャップをより早い段階で統計的に検出する分析技術を開発した。また，PCヘルスモニタリングの情報を活用し，より高いレベルで顧客満足を実現する品質保全サービスの検討も進めている（同，p.12－15参照）。

表2. 東芝の取組みと製品ライフサイクル高信頼化モデルとの対応
Classification of technologies developed by Toshiba in product life-cycle quality assurance models

東芝の取組み	対象とするギャップ				提供する技術		
	仕様と実装のギャップ	実装と環境のギャップ	環境と仕様のギャップ	共通	ギャップ監視支援	ギャップ検出及び分析支援	ギャップ修正及び緩和支援
デジタル機器ヘルスモニタリング	○		○		○	○	○
プロアクティブ品質保全	○				○	○	○
ネットワーク接続診断			○			○	○
プログラム部品検証	○					○	
テストケース自動生成	○					○	
電子機器高品位設計		○				○	
業務プロセスリスク評価と改善			○		○	○	○
障害情報フィルタリング				○	○	○	
遠隔監視プラットフォーム				○	○	○	○

○：対応あり

■ネットワーク接続の診断

PCを，技術に詳しくない一般ユーザーが購入し使用する場合が増えている。一般ユーザーが無線LANの接続に関してトラブルを抱えた際に，ネットワーク接続状況を自動で診断し，サポートセンターから適切な指示を短時間で得られることを目的とした，無線LAN診断ツールを開発した。仕様とユーザーの使用環境のギャップを検出し，ユーザーによる修正を支援する技術である（同，p.16－19参照）。

■プログラム部品の検証

仕様と実装（プログラム）を関係論理と呼ばれる論理で表現し，実装が仕様を満たしているかを検証する技術を開発した。C言語のようにデータ構造とポインタ演算を持つプログラミング言語の検証ができる点が特長である。C言語プログラム部品を仕様と実装の対で管理し，ギャップを検出することで，製品設計・製造時だけでなく，製品の更新・拡張時の再利用部品の信頼性を保証できる（同，p.20－23参照）。

■テストケースの自動生成

対象システムのモデルから網羅的な検証を可能とするテストケース自動生成方式を搭載したモデルベーステスト技術を開発した。

システムの状態を適切に同一視して状態遷移系を縮約することで，従来技術よりも効率的なテストケースを生成できる点が特長である。モデル化にはコストを要するが，派生開発におけるシステムの改変・拡張時の仕様と実装のギャップを，テストの自動化により効率よく検出できる（同，p.24－27参照）。

■プラットフォームの高品位設計

組込みアプリケーションプログラムが稼働するHWとSWのプラットフォームの設計環境を開発した。CPU，メモリ，及びバスの構成と構造を自由に変更したり合成できるESL（Electric System Level Design）ツール S.E.R. studio で作成したモデルを，システムシミュレータを使って解析し，HWとSWのプラットフォームの性能や品質を短期間で分析し評価する。製品プロダクトラインの

(注3)　プロアクティブ
将来を予測し事前に対策を取ること。

派生開発におけるアプリケーションプログラムの変更及びHWとSWのプラットフォームの変更に対して，早い段階で性能や品質の検証を行うことができる。すなわち，実装としてのアプリケーションプログラムとそれが稼働する環境としてのプラットフォームのギャップを検出することができる（同，p.28－31参照）。

■業務プロセスリスクの評価と改善

医療業務をモチーフに，小さい確率であっても許容範囲を超える損失が発生する可能性があるかどうかを定量的に評価できるRFMEA（Risk FMEA）手法と，リスクが業務プロセスのどこで発生しているかを可視化するインシデント レポート収集システムを開発した。

この手法により，システムの運用時に，業務プロセスの仕様としての想定リスクと環境としての実際のリスクのギャップをPDCA（Plan-Do-Check-Act）サイクルで改善し，リスクを低減できる（同，p.32－36参照）。

■障害情報のフィルタリング

障害が各所で同時に発生することのある大規模なシステムを監視する場合，1次的な障害回避後に保守員による機能復旧や事後対策を迅速に行うために，保守員が障害情報を的確に認識する必要がある。しかし，現実には多くのサイトから収集される障害情報は膨大となり，人間による情報監視の限界を超えてしまう。また，保守員にとって情報監視ノウハウの継承も課題となっている。

そこで，大量の障害情報の中から，各保守員が真に認識すべき情報を抽出する障害情報フィルタリング技術を開発した。ギャップ監視の重要技術の一つである（同，p.37－40参照）。

■遠隔監視のプラットフォーム

ビルや，ガス，水道，道路，空港などの社会インフラシステム分野で長年培ったインターネットを利用した“遠隔監視”の技術と，保守に関する様々なデータを一元管理する“保守統合データベース”を連携することにより，“何かあったときの保守”だけでなく，“情報を蓄積し分析することによる先進的で最適な保守”を実現するTMSTATION$_{TM}$を開発した。これは，ギャップ監視をベースに製品ライフサイクル高信頼化を支える汎用のプラットフォームである（同，p.41－44参照）。

製品ライフサイクル高信頼化の今後の展望

製品の仕様，実装，及び環境の変化によって生じるギャップに対して，監視，検出，修正により製品ライフサイクル高信頼化を実現する技術を示し，当社の取組みについて述べた。

今後，ギャップの監視や検出・分析技術が洗練化され，実用化が進むとともに，次のチャレンジである運用・実行時におけるギャップの自動修正・緩和に関する研究開発が進むと思われる。過去の実行で実績のある挙動を“暗黙の仕様”として抽出し，現在の実装のタイミングなど挙動に活用する技術は，その一例である[9]。

当社は，ここで述べた個々の技術を洗練化し展開を図るとともに，これらを体系的に整備し連携させることで，SW，HW，そして人間系プロセスを含めたシステム全体の製品ライフサイクル高信頼化の技術開発を推進し，時代の要請に応える品質のイノベーションを実現していく。

文　献

(1) 日本学術会議 情報学委員会 セキュリティ・ディペンダビリティ分科会．“提言：安全・安心を実現する情報社会基盤の普及に向けて”．<http://www.scj.go.jp/ja/info/kohyo/pdf/kohyo-20-t58-4.pdf>，（参照 2009-07-10）．

(2) Leveson, N. G. "System safety engineering : back to the future". <http://sunnyday. mit. edu/book2.pdf> , (accessed 2009-07-10)

(3) 内平直志．製造業のサービスイノベーションのための知識処理技術．人工知能学会誌．**22**, 6, 2007, p.754－762.

(4) Michael G. Pecht. Prognostics and Health Management of Electronics. Wiley & Blackwell, 2008, 316p.

(5) 池田信之，ほか．モデル検査によるソフトウェア上流設計の品質向上技術．東芝レビュー．**64**, 4, 2009, p.40－43.

(6) Rushby, J. Runtime Certification, RV2008, Lecture Notes in Computer Science. 5289, Springer-Verlag, 2008, p.21－35.

(7) 進　博正，ほか．ソフトウェアの動作検証支援システム ARVE．東芝レビュー．**62**, 9, 2007, p.55－58.

(8) Ernst, M. D., et al. The Daikon system for dynamic detection of likely invariants. Sci. Comput. Program. **69**, 1-3, 2007, p.35－45.

(9) Uchihira, N. Making Reactive Systems Highly Reliable by Hypersequential Programming. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **E88-A**, 4, 2005, p.941－947.

内平　直志
UCHIHIRA Naoshi, D.Eng.
研究開発センター次長（システム・基盤領域担当），工博。ソフトウェア工学，リスク工学，サービス科学の研究・開発に従事。電子情報通信学会会員。
Corporate Research & Development Center